

Bounded Multivariate Surfaces On Monovariate Internal Functions

Shriprakash Sinha and Gert J. ter Horst*

Neuroimaging Center, Antonius Deusinglaan 2
University Medical Center Groningen, 9713 AW, Groningen, The Netherlands
{shriprakash.sinha@gmail.com, g.j.ter.horst@med.umcg.nl}

Abstract. Combining the properties of monovariate internal functions as proposed in Kolmogorov superimposition theorem, in tandem with the bounds wielded by the multivariate formulation of Chebyshev inequality, a hybrid model is presented, that decomposes images into homogeneous probabilistically bounded multivariate surfaces. Given an image, the model shows a novel way of working on reduced image representation while processing and capturing the interaction among the multidimensional information that describes the content of the same. Further, it tackles the practical issues of preventing leakage by bounding the growth of surface and reducing the problem sample size. The model if used, also sheds light on how the Chebyshev parameter relates to the number of pixels and the dimensionality of the feature space that associates with a pixel. Initial segmentation results on the Berkeley image segmentation benchmark indicate the effectiveness of the proposed decomposition algorithm.

Key words: Multivariate Inequality, Monovariate functions, Bounded Surfaces

1 Introduction

In order for an image to be decomposed, a proper representation of the image must first be done. One set of solutions for image representation is the decomposition of multivariate functions into monovariate functions as proposed by Kolmogorov superimposition theorem (KST) [1]. Sprecher et.al [2] have also proved that the monovariate internal functions obtained via the KST can be used to build space filling curves that sweep a multidimensional space. These 1D representations of the image can then be exploited for further processing using simple univariate or bivariate signal processing methods, as has been shown in

* This work was conducted as a part of ongoing research in segmenting and labeling brain images at the Neuroimaging Center, in University Medical Center Groningen, The Netherlands. Shriprakash Sinha is with the Neuroimaging Center at University Medical Center Groningen, Antonius Deusinglaan 2, 9713 AW, Groningen, The Netherlands. Dr. Gert J. ter Horst is the General Director of the Neuroimaging Center, UMCG.

[3] and [4]. It has further been proposed in [4] that either the space filling curves can be fixed and then construct external function whose sum and compositions correspond to a multivariate function [2] or produce an algorithm that generates the internal function that adapts to the multivariate function and gives different space filling curve for different multivariate functions [5]. The work in this manuscript finds its motivation in presenting an initial hybrid model that uses a fixed space filling curve for an image and differs in employing multivariate Chebyshev inequality on the \mathcal{N} dimensional points lying on the curve, to yield homogeneous probabilistically bounded multivariate surfaces.

From the theory of space filling curves, it is known that the Hilbert Space Filling Curve (HSFC) [6] is the best in preserving the clustering properties while taking into account the locality of objects in a multidimensional space ([7], [8]). Even though it can be applied to transform a multidimensional image representation into a linear format ([3], [4]), the manuscript applies the HSFC to transform a 2D matrix into 1D space filling curve. The reason being that it saves the time and avoids the complexity in processing a 2D matrix in comparison to $\mathcal{N}D$ matrix.

Next, a multivariate formulation of the generalized Chebyshev inequality [9] is applied to decompose the image into surfaces bounded probabilistically via a single Chebyshev parameter. Since the bounded surfaces are constructed based on the interaction of a set of points in $\mathcal{R}^{\mathcal{N}}$ lying on the curve, it can be safely assumed that information about the nature or density of surface in a locality gets captured in these tiny patches. For example, RGB images from the Berkeley Segmentation Benchmark (BSB) [10] have been taken into consideration for the current study. In the case of a single RGB image, three dimensions exist in the colour map. These three form a feature set ($\mathcal{N} = 3$).

Several advantages arise with the use of this new hybrid model, namely: • Faster processing of the image on 1D compared to analysis of neighbourhood information per pixel. • Generation of homogeneous surfaces of different sizes that are bounded probabilistically, by inequality. • Leakage problem gets avoided due to conservative nature of inequality. • Reduction in problem sample size by a factor ϵ (Chebyshev parameter). • The density of image gets investigated via the multivariate formulation of the inequality. • The generalized hybrid model adapts to multivariate information while traversing on a 1D fixed curve.

Hitherto, a brief description of the state of the work has been covered. In section 2, the theoretical aspect of the method is dealt in a greater detail with a toy image example. Experiments section 4 deals with the empirical evaluations conducted on BSB. Lastly, the conclusion follows in section 6.

2 Theoretical Perspective

2.1 Hilbert Space Filling Curve

The space filling curves form an important subject as it helps in transforming a multidimensional dataset into a linear format. This comes at a price of losing

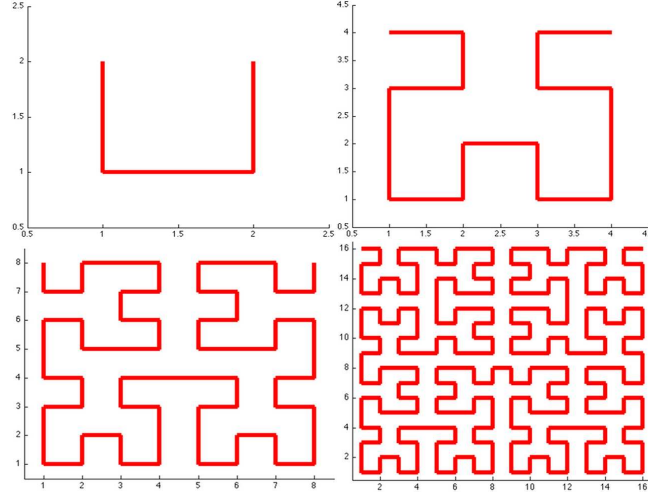


Fig. 1. Hilbert Space Filling Curve for grid of size 2 (top left), 4 (top right), 8 (bottom left) and 16 (bottom right).

some amount of information, but the merits of preserving the local properties while transforming the objects in multi dimension to single dimension outweigh the incurred cost. The HSFC is a fractal filling curve proposed by [6] which fills the space of 2D plane in a continuous manner. Analytical results found in [8], [7] and [16] prove the optimality of results obtained while using HSFC. In the current formulation a matlab implementation of [17] is used to generate the HSFC for 2D matrices.

Figure 1 shows the space filling curve for the grids of size 2, 4, 8 and 16 respectively. Note that the curve covers each and every point on the integer grid once while taking into account the local properties. It is not that the HSFC does not work for rectangular matrices, but the analysis of cluster preserving properties of the same becomes asymptotic in nature rather than being exact, as has been proved in [8]. Given an \mathcal{N} D image, the HSFC is generated which remains invariant of the same.

2.2 Multivariate Chebyshev Inequality

Let X be a stochastic variable in \mathcal{N} dimensions with a mean $E[X]$. Further, Σ be the covariance matrix of all observations, each containing \mathcal{N} features and $\epsilon \in \mathcal{R}$, then the multivariate TChebyshev Inequality in [9] states that:

$$\begin{aligned} \mathcal{P}\{(X - E[X])^T \Sigma^{-1} (X - E[X]) \geq \epsilon\} &\leq \frac{\mathcal{N}}{\epsilon} \\ \mathcal{P}\{(X - E[X])^T \Sigma^{-1} (X - E[X]) < \epsilon\} &\geq 1 - \frac{\mathcal{N}}{\epsilon} \end{aligned} \tag{1}$$

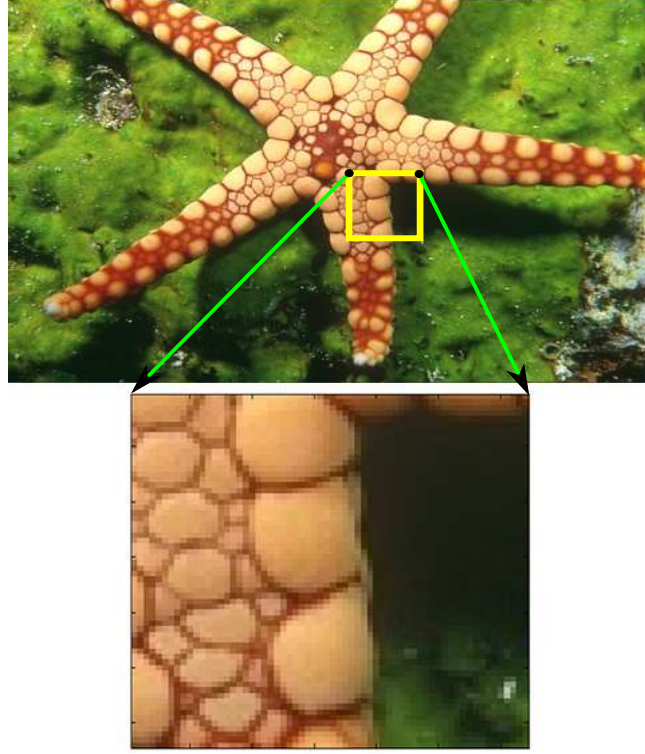


Fig. 2. Starfish image from [10] and the 64×64 block under consideration.

i.e. the probability of the spread of the value of X around the sample mean $E[X]$ being greater than ϵ , is less than \mathcal{N}/ϵ . There is a minor variation for the univariate case stating that the probability of the spread of the value of x around the mean μ being greater than $\epsilon\sigma$ is less than $1/\epsilon^2$. Apart from the minor difference, both formulations convey the same message about the probabilistic bound imposed when a random vector or number X lies outside the mean of the sample by a value of ϵ .

In a broader perspective, the goal being to demarcate regions based on surfaces, two questions need to be addressed regarding the decomposition of image.

- Which two pixels or their corresponding \mathcal{ND} vectors be selected to initialize a surface depicting *near uniform behaviour*?
- What should be the size of such a restricted surface?

2.3 Initializing Surface

The solution to first question would help in *initializing a surface*. A pair of vectors in \mathcal{ND} will swap a flat plane with an angle subtended in between the

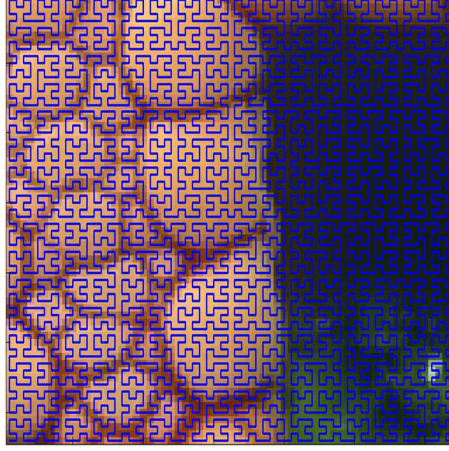


Fig. 3. HSFC on the patch of 64×64 .

two vectors. Given that the dot product exists in the higher dimensional plane, the cosine of the angle θ between the two vectors would suggest the degree of closeness between them. If \mathbf{a} and \mathbf{b} are two such vectors in \mathcal{ND} , then the degree of closeness is given by:

$$\cosine\theta = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|_2 \times \|\mathbf{b}\|_2} \quad (2)$$

were, $\langle \mathbf{a}, \mathbf{b} \rangle$ is the dot product and the denominator contains the 2-norm terms of both vectors. It is well known that the absolute value of $\cosine\theta$ tends to 1 (0) as vectors tend to be nearly parallel (perpendicular). Let \mathbf{u} , \mathbf{v} and \mathbf{w} be three consecutive pixels on the HSFC. If the *cosine* of the angle between \mathbf{u} and \mathbf{v} evaluates to an absolute value greater than a nearness threshold $Npar$ (say 0.95) then the pair is considered as a valid surface. Note that $Npar$ is the nearness parameter which is used as a threshold to decide the degree of closeness of two pixels for forming a surface. If not, then \mathbf{u} is left as a single point in \mathcal{ND} and the closeness criterion is checked for \mathbf{v} and \mathbf{w} (and the process is repeated).

2.4 Size Of Surface

Solving the second question shall define the *range of the surface*. Once the start and end points (say \mathbf{v} and \mathbf{w} respectively) of a valid surface have been set, the size of the surface has to be determined. The size of the surface would constitute all points that contribute towards uniform surface behaviour in \mathcal{ND} . This degree of uniformity is controlled via the Chebyshev's Inequality. The idea is executed as follows: The next consecutive point (say \mathbf{t}) after \mathbf{w} is considered for surface extent analysis. If the spread of surface point \mathbf{t} from $E(surface)$ the mean of

the *existing surface* $[\mathbf{v}, \mathbf{w}]$, factored by the covariance matrix, is below ϵ , then \mathbf{t} is considered as a part of the surface and marked as a new ending point of the surface. Using Chebyshev's Inequality, it boils down to:

$$\begin{aligned} \mathcal{P}\{(\mathbf{t} - E([\mathbf{v}, \mathbf{w}]))^T \Sigma^{-1} (\mathbf{t} - E([\mathbf{v}, \mathbf{w}])) \geq \epsilon\} &\leq \frac{\mathcal{N}}{\epsilon} \\ \mathcal{P}\{(\mathbf{t} - E([\mathbf{v}, \mathbf{w}]))^T \Sigma^{-1} (\mathbf{t} - E([\mathbf{v}, \mathbf{w}])) < \epsilon\} &\geq 1 - \frac{\mathcal{N}}{\epsilon} \end{aligned} \quad (3)$$

where Σ is the covariance matrix between the $\mathcal{N}\mathcal{D}$ vectors constituting the initial surface. Satisfaction of this criterion leads to extension of the size of initial surface by one more point i.e. \mathbf{t} . The surface now constitutes $[\mathbf{v}, \mathbf{w}, \mathbf{t}]$, with \mathbf{v} and \mathbf{t} as start and end marker points. If not, the size of the surface remains as it is and a fresh start is made starting with \mathbf{t} and the next consecutive point on the HSFC. The satisfaction of the inequality also gives a lower probabilistic bound on size of surface by a value of $1 - (\mathcal{N}/\epsilon)$, if the second version of the Chebyshev formula is under consideration.

The above formulation implies that when a homogeneous patch is encountered, then the a new point does not deviate much from the initial surface. Thus the size of the surface grows smoothly. For a highly irregular patch, the surface size may be very restricted due to high variation of a pixel from the surface it is being tested in the vicinity. Figure 2 shows the tiny patch (64×64) of starfish under consideration and figure 3 shows the HSFC generated over the area of the image.

The tiny restricted surfaces generated using the multivariate and the univariate formulation of the Tchebyshev inequality for $\epsilon = 3$ are shown in figure 4 and 5. Note how the surfaces differ due to the multivariate and univariate formulation of the inequality. The former takes into account the entire $\mathcal{N}\mathcal{D}$ vectors in tandem to compute the covariance or the texture interaction and the mean, while the later computes the inequality separately for each and every dimension. The different colours just indicate the different surfaces and has nothing to do with clustering at this stage. The potentiality of the method gets highlighted due to the bounded surfaces that have been obtained after image decomposition. This boundedness is checked via the parameter which determines the degree of intricacy to which the texture interaction is to be taken into account. It should be noted that in the univariate case the decision is made on the majority vote after \mathcal{N} evaluations of the inequality on the space filling curve for a single pixel. What paves way is that the univariate formulation does not capture the interaction which later leads to low grade segmentation results compared to that given by the formulation. These differences are apparent in the figures mentioned at the starting of the paragraph. Figure 6 and 7 shows the image patch decomposed into surface patches for different ϵ values.

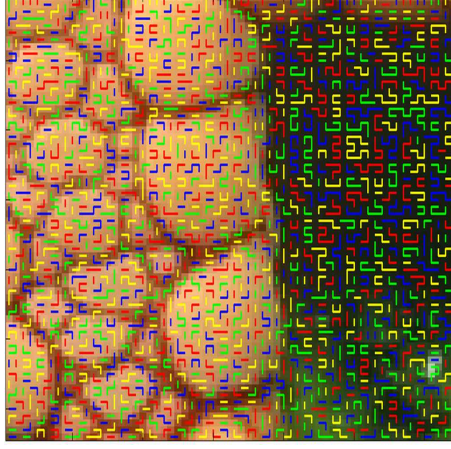


Fig. 4. Patch of 64×64 image of starfish from [10], decomposed using the multivariate formulation of Chebyshev inequality with $\epsilon = 3$ and $N_{par} = 0.95$. A coloured line shows the pixels associated to a single bounded surface on the HSFC.

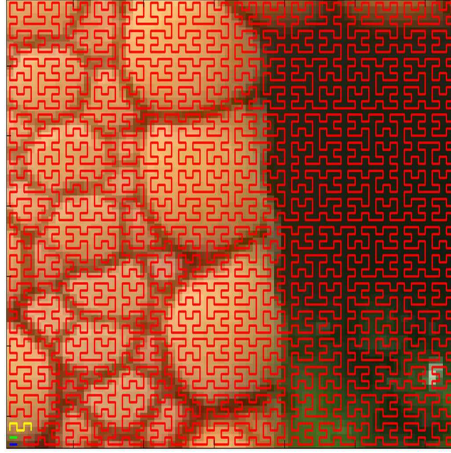


Fig. 5. Patch of 64×64 decomposed using the univariate formulation of Chebyshev inequality with $\epsilon = 3$ and $N_{par} = 0.95$. A coloured line shows the pixels associated to a single bounded surface on the HSFC. Note that size of surface is decided based on voting across \mathcal{N} evaluations.

2.5 Implications

The inequality being a criterion, the probability associated with the same gives a belief based bound on the satisfaction of the criterion. This gives rise to certain

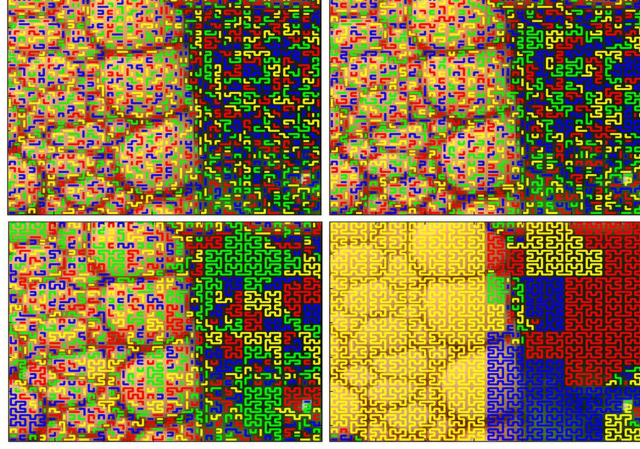


Fig. 6. Decomposed surfaces using multivariate inequality with $N_{par} = 0.95$ and for ϵ equal to 4 (top left), 8 (top right), 16 (bottom left) and 32 (bottom right).

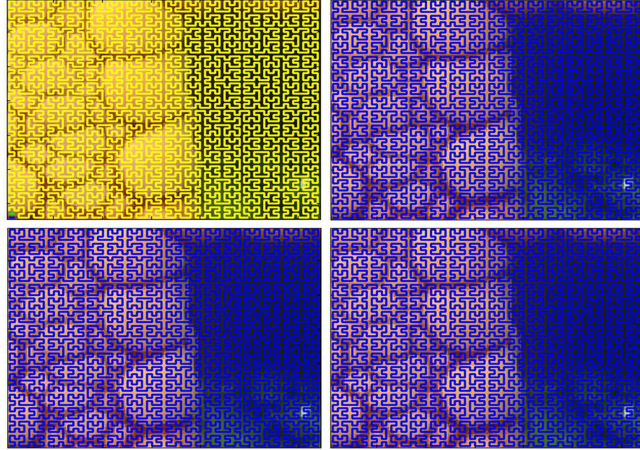


Fig. 7. Decomposed surfaces using univariate inequality with $N_{par} = 0.95$ and for ϵ equal to 4 (top left), 8 (top right), 16 (bottom left) and 32 (bottom right).

simple implications as follows. Let \mathcal{D} be a *decomposition* which is equivalent to $(X_t - E[X])^T \Sigma^{-1} (X_t - E[X])$. Then:

Lemma 1. *Decompositions \mathcal{D} are bounded by lower probability bound of $1 - (\mathcal{N}/\epsilon)$ given that $\epsilon \geq \mathcal{N}$.*

Not only does it hold true for $\epsilon \geq \mathcal{N}$ but also for $\epsilon < \mathcal{N}$. But the probability being greater than a negative value is always true and thus $\epsilon = \mathcal{N}$ forms the lower bound. As $\epsilon \rightarrow \infty$, $\mathcal{P}(\mathcal{D} < \epsilon) \rightarrow 1$.

Lemma 2. *The value of ϵ reduces the size of the sample from \mathcal{M} to an upper bound of \mathcal{M}/ϵ probabilistically with a lower bound of $1 - (\mathcal{N}/\epsilon)$. Here \mathcal{M} is the number of pixels in a 2D matrix.*

This holds true as the image is decomposed into surfaces which are probabilistically bounded via the Chebyshev inequality. This decomposition leads to reduction in the sample size by a factor of ϵ while retaining the information content, kudos to the space filling curve traversal.

Lemma 3. *As $\epsilon \rightarrow \mathcal{N}$ the lower probability bound drops to zero, implying large number of small decompositions \mathcal{D} can be achieved. (Vice versa for $\epsilon \rightarrow \infty$)*

This gives an insight into the degree to which the image can be decomposed. Where finer details are of import, one may use values of ϵ tending to \mathcal{N} and vice versa.

Theorem 1. *Let image \mathcal{I} contain \mathcal{M} pixels, with each pixel having \mathcal{N} features. If \mathcal{I} can be decomposed in $\ell = \mathcal{M}/\epsilon$ bounded surfaces via the proposed hybrid model, then in case of the decompositions having equally likely probabilities: (a) $\mathcal{M} = \epsilon^2/(\epsilon - \mathcal{N})$ and (b) $\epsilon \in$ open interval $(\mathcal{N}, \mathcal{M})$.*

Proof. Since the \mathcal{I} can be decomposed into ℓ bounded surfaces, it is known that the decompositions are *disjoint* sets. Let \mathcal{D}_j (for $j \in \{1, \ell\}$) be such decompositions. Then $\mathcal{I} = \bigcup_{j=1}^{\ell} \mathcal{D}_j$. Considering \mathcal{I} as the universal set, we get: $\mathcal{P}(\mathcal{I}) = \mathcal{P}(\bigcup_{j=1}^{\ell} \mathcal{D}_j) = \sum_{j=1}^{\ell} \mathcal{P}(\mathcal{D}_j)$. From lemma 1, it is known that $\mathcal{P}(\mathcal{D}_j < \epsilon) \geq 1 - (\mathcal{N}/\epsilon)$. In the case that the decompositions are equally likely, the lowest probability for each \mathcal{D}_j evaluates to $1 - (\mathcal{N}/\epsilon)$.

Thus, $\sum_{j=1}^{\ell} \mathcal{P}(\mathcal{D}_j) = \ell(1 - \mathcal{N}/\epsilon) = (\mathcal{M}/\epsilon)(1 - \mathcal{N}/\epsilon)$. Since $\mathcal{P}(\mathcal{I}) = 1$ and $\mathcal{P}(\mathcal{I}) = \sum_{j=1}^{\ell} \mathcal{P}(\mathcal{D}_j)$, it implies that $(\mathcal{M}/\epsilon)(1 - \mathcal{N}/\epsilon) = 1$. Simplifying the foregoing formulation leads to $\mathcal{M} = \epsilon^2/(\epsilon - \mathcal{N})$. This proves the part (a) of the theorem.

From part (a), it can be clearly seen that $\epsilon - \mathcal{N} \neq 0$, lest part (a) would be invalid. Thus $\epsilon \neq \mathcal{N}$. Similarly, if $\epsilon = \mathcal{M}$, then on simplification of part (a) evaluates to $\mathcal{N} = 0$. Again this cannot be the case as an image will have atleast one feature in terms of intensity. Thus $\epsilon \neq \mathcal{M}$. Let $k > 1$, then for $\epsilon = \mathcal{N}/k$ part (a) shows that the ratio of \mathcal{M}/\mathcal{N} is a negative quantity. This cannot be possible as both numbers in the ratio are positive numbers. Thus $\epsilon > \mathcal{N}$. Again, for the same condition of k , if $\epsilon = k\mathcal{M}$, part (a) on evaluation shows the ratio of \mathcal{M}/\mathcal{N} to be a negative quantity. This again is a contradiction given the state of \mathcal{M} and \mathcal{N} . Thus $\epsilon < \mathcal{M}$. Thus ϵ is strictly bounded in the open interval $(\mathcal{N}, \mathcal{M})$. This finishes proof for part (b). \square

Theorem 2. *If an image \mathcal{I} is decomposed into bounded surfaces s.t. the latter have equally likely probabilities, then theoretically the maximum value of both ϵ (Chebyshev parameter) and \mathcal{N} (number of feature dimensionality of \mathcal{I}) is of the order of $\sqrt{\mathcal{M}}$.*

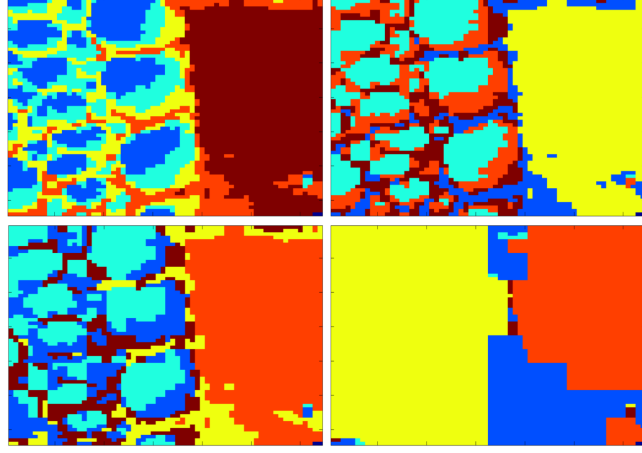


Fig. 8. Image with 5 clusters using k -means (cityblock distance, 100 replicates and 1000 iterations) on decomposed surfaces generated via multivariate inequality with $N_{par} = 0.95$ and for ϵ equal to 4 (top left), 8 (top right), 16 (bottom left) and 32 (bottom right).

Proof. From theorem 1, it is known that $\mathcal{M} = \epsilon^2/(\epsilon - \mathcal{N})$, when the probabilities of the decomposed surfaces are equally likely. It is also known that $\mathcal{N} < \epsilon < \mathcal{M}$. Let $k > 1$ and ϵ decrease harmonically via $\epsilon = \mathcal{M}/k$ for k from 2 onwards. Then simplifying part (a) of theorem 1 gives $\mathcal{M}/\mathcal{N} = k^2/(k - 1)$.

Now, if $\epsilon = \mathcal{M}/\mathcal{N}$, then $\mathcal{M}/k = k^2/(k - 1)$. The fraction on the right hand side can be segregated into complete and partial fractions as $k + k/(k - 1)$. It is known that $k + 1 < k + k/(k - 1) < k + 2$. Then $k + 1 < \mathcal{M}/k < k + 2$. Equating for both inequalities around \mathcal{M}/k , the value of k lies between $\sqrt{(1 + \mathcal{M})} + 1$ and $(\sqrt{(1 + 4\mathcal{M})} - 1)/2$. Taking the order of maximum value of k as $\sqrt{(\mathcal{M})}$, $\epsilon = \mathcal{M}/k = \sqrt{(\mathcal{M})}$ and $\mathcal{N} = k = \sqrt{(\mathcal{M})}$. \square

Theorem 1 shows how the sample size \mathcal{M} is related to the dimensionality of feature space \mathcal{N} via the Chebyshev parameter ϵ . Implicitly, it also states that the sample size \mathcal{M} must be greater than \mathcal{N} and the value of ϵ lies in the open interval $(\mathcal{N}, \mathcal{M})$. These tight bounds in an idealistic case show that the model is effective in a theoretical sense. The second theorem builds on the first and shows that the maximum theoretical value of the parameter ϵ is of the order of square root of the sample size \mathcal{M} and so is the dimensionality of the feature space \mathcal{N} . In an ideal case of equal likelihood, this maximum value gives an upper bound on ϵ as well as \mathcal{N} , such that the decompositions are uniformly spread in the higher dimensional space.

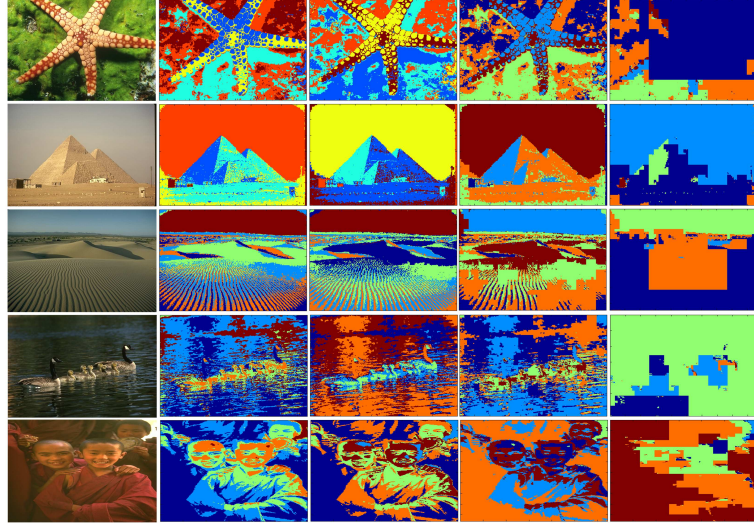


Fig. 9. Images from [10] segmented for $N_{par} = 0.95$ and for ϵ equal to 4, 8, 16 and 32, from left but one to right with number of clusters 5.

2.6 Clustering On Bounded surfaces

Once the image has been decomposed into bounded surfaces, segmentation of the image is done on the average value of surfaces via k -means algorithm in Matlab with a certain number of pre-defined clusters. The cityblock distance is used as a metric for the kmeans and the number of replicates is of the order of 100 with 1000 iterations for the k -means. The reason for using a high number replicates and iterations is to avoid getting stuck in local solutions. The average values are computed by taking the mean of the N_D points that constitute the bounded surfaces. These values are considered to be robust as the surfaces themselves are bounded on the first place probabilistically taking into account the variability in intensity behaviour.

Figure 8 shows the corresponding result for different values of ϵ . The figure shows that the quality of segmentation degrades as the value of ϵ increases, which increases the surface size. For example, the groves on the starfish are captured in greater details for $\epsilon = 4$ than for higher values of the same. A few sample images from [10] for which segmented images were generated over different values of $\epsilon \in \{4, 8, 16, 32\}$ have been presented in figures 9 and 10. The number of clusters was predefined to be 5. With increasing size of ϵ values, the amount of decompositions reduce which later affect the quality of the segmentation. This is apparent in the figures as one moves from left to right. The clustering on these bounded surfaces is not only good but also less time consuming as the segmentation are done on a reduced sample size while still retaining the crucial

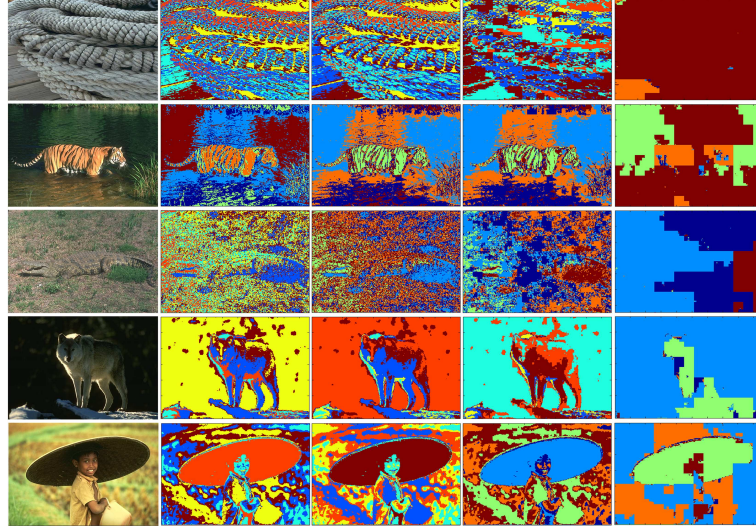


Fig. 10. Images from [10] segmented for $N_{par} = 0.95$ and for ϵ equal to 4, 8, 16 and 32, from left but one to right with number of clusters 5.

pieces of information. Not that the solution is the best, but results tend to be good quality at first sight.

3 Decomposition Algorithm

Algorithm 1 and its continuity in 2 shows the implementation for the decomposing the image into probabilistically bounded surfaces based on space filling curve traversal. The depicted version is for multivariate formulation of the Tchebyshev inequality. Minor change in the form of univariate formulation used separately with each of the \mathcal{N} dimensions would lead to univariate version.

Note that the output of the decomposition algorithm is a list of bounded surfaces. Many features could be developed but in this manuscript mean values of all the $\mathcal{N}D$ points constituting a surface is taken as a feature vector. This is because the inequality measures the degree of homogeneity of density interaction in $\mathcal{N}D$.

Another point to be aware of is the computation of the measure in multivariate Chebyshev inequality in equation 3. Since it requires presence of inverse of the covariance matrix, one may run into problem of sparseness, or inappropriate dimensionality between the number of samples and features. To overcome these problems, the pseudo inverse was computed using the singular value decomposition implementation in matlab. Being fast and effective, the decomposition of the image into bounded surfaces works within a matter of seconds.

Algorithm 1 Tchebyshev Surfaces

```

1: procedure TCHSURF( $img, \epsilon, Npar, formulation$ )
2:    $[nrows, ncols] \leftarrow \text{size}(img)$ 
3:    $[R, S] \leftarrow \text{hlbrtrcv}(nrows, ncols)$  ▷ HSFC coords
   Initialize Variables
4:    $marked_{vertex} \leftarrow []$  ▷ Marked pixels per surface
5:    $surface_{list} \leftarrow []$  ▷ List of surfaces
6:    $feature_{list} \leftarrow []$  ▷ List of features per surface
7:    $surface_{no} \leftarrow 0$  ▷ Count of surfaces
   Generate bounded surfaces on HSFC
8:    $s_{idx} \leftarrow 1$  ▷ Index of pt where the curve starts
9:   while  $s_{idx} < len - 1$  do ▷  $len$  is length of curve
10:     $e_{idx} \leftarrow s_{idx} + 1$ 
11:     $[rs, cs] \leftarrow [R(s_{idx}), S(s_{idx})]$ 
12:     $[re, ce] \leftarrow [R(e_{idx}), S(e_{idx})]$ 
13:     $pix_{info} \leftarrow \text{struct}()$ ; ▷ Initially two pixels
14:     $pix_{info}.loc \leftarrow [rs, cs; re, ce]$  ▷ store locations
   Store intensity values per pixel
15:     $surface \leftarrow []$ 
16:     $[lenidx, cols] \leftarrow \text{size}(pix_{info}.loc)$ 
17:    for  $i = 1 : lenidx$  do
18:       $[r, c] \leftarrow [pix_{info}.loc(i, 1), pix_{info}.loc(i, 2)]$ 
19:       $temp \leftarrow []$ 
20:      for  $j = 1 : \mathcal{N}$  do ▷ Features
21:         $temp \leftarrow [temp; img(c, r, j)]$ 
22:      end for
23:       $surface \leftarrow [surface, temp]$ 
24:    end for
   Compute nearness between two initial pixels
25:     $innerprod \leftarrow \text{dot}(surface(:, 1), surface(:, 2));$ 
26:     $cosval \leftarrow \frac{innerprod}{\text{norm}(surface(:, 1), 2) \times \text{norm}(surface(:, 2), 2)}$ 
   Check  $\cosine\theta$  greater than nearness param
27:    if  $cosval \geq Npar$  then
28:       $e_{idx} \leftarrow e_{idx} + 1$ 
29:      while  $e_{idx} \leq len$  do
   Store next pixel on HSFC
30:         $[re, ce] \leftarrow [R(e_{idx}), S(e_{idx})]$ 
31:         $pix_{loc}^{end} \leftarrow [re, ce]$ 
   Store intensity values for pixel
32:         $new_{pix} \leftarrow []$ 
33:        for  $j = 1 : \mathcal{N}$  do
34:           $new_{pix} \leftarrow [new_{pix}; img(ce, re, j)]$ 
35:        end for

```

Algorithm 2 Tchebyshev Surfaces Continued

Tchebyshevs inequality for fixing length of surf
 NOTE - rows are dimension and cols are pixels in surface matrix

```

36:       $surfacemu \leftarrow \mu(surface^T)$ 
37:       $surfacestd \leftarrow std(surface^T)$ 
38:       $covR \leftarrow cov(surface^T)$ 
39:       $dev \leftarrow (new_{pix} - surfacemu^T)$ 
  Compute inequality in equ 3
40:       $criterion \leftarrow dev^T * pinv(covR) * dev$ 
41:       $decision \leftarrow 0$ 
42:      if  $criterion < \epsilon$  then
43:           $decision \leftarrow decision + 1$ 
44:      end if
45:      if  $decision \geq 1$  then
46:           $surface \leftarrow [surface, new_{pix}]$ 
47:           $pix_{info}.loc \leftarrow [pix_{info}.loc; pix_{loc}^{end}]$ 
48:           $e_{idx} \leftarrow e_{idx} + 1$ 
49:      else
50:           $break$ 
51:      end if
52:  end while
  Store the surfaces, associated pixels and features
53:       $surface_{no} \leftarrow surface_{no} + 1$ 
54:       $surface_{list}surface_{no} \leftarrow pix_{info}.loc$ 
  Mean per dimension as feature for a surface
55:       $featval \leftarrow []$ 
56:      for  $i = 1 : \mathcal{N}$  do
57:           $featval \leftarrow [featval; \mu(surface(i, :))]$ 
58:      end for
59:       $feature_{list}\{surface_{no}\} \leftarrow featval$ 
60:       $s_{idx} \leftarrow e_{idx}$ 
61:  else ▷ If only one pixel is a surface
62:       $pix_{info}.loc(2, :) \leftarrow []$ 
63:       $surface(:, 2) \leftarrow []$ 
64:       $surface_{no} \leftarrow surface_{no} + 1$ 
65:       $surface_{list}\{surface_{no}\} \leftarrow pix_{info}.loc$ 
  Mean per dimension as a feature for just one pixel
66:       $featval \leftarrow []$ 
67:      for  $i = 1 : \mathcal{N}$  do
68:           $featval \leftarrow [featval; \mu(surface(i, :))]$ 
69:      end for
70:       $feature_{list}\{surface_{no}\} \leftarrow featval$ 
71:       $s_{idx} \leftarrow s_{idx} + 1$ 
72:  end if
73:  end while
74: end procedure

```

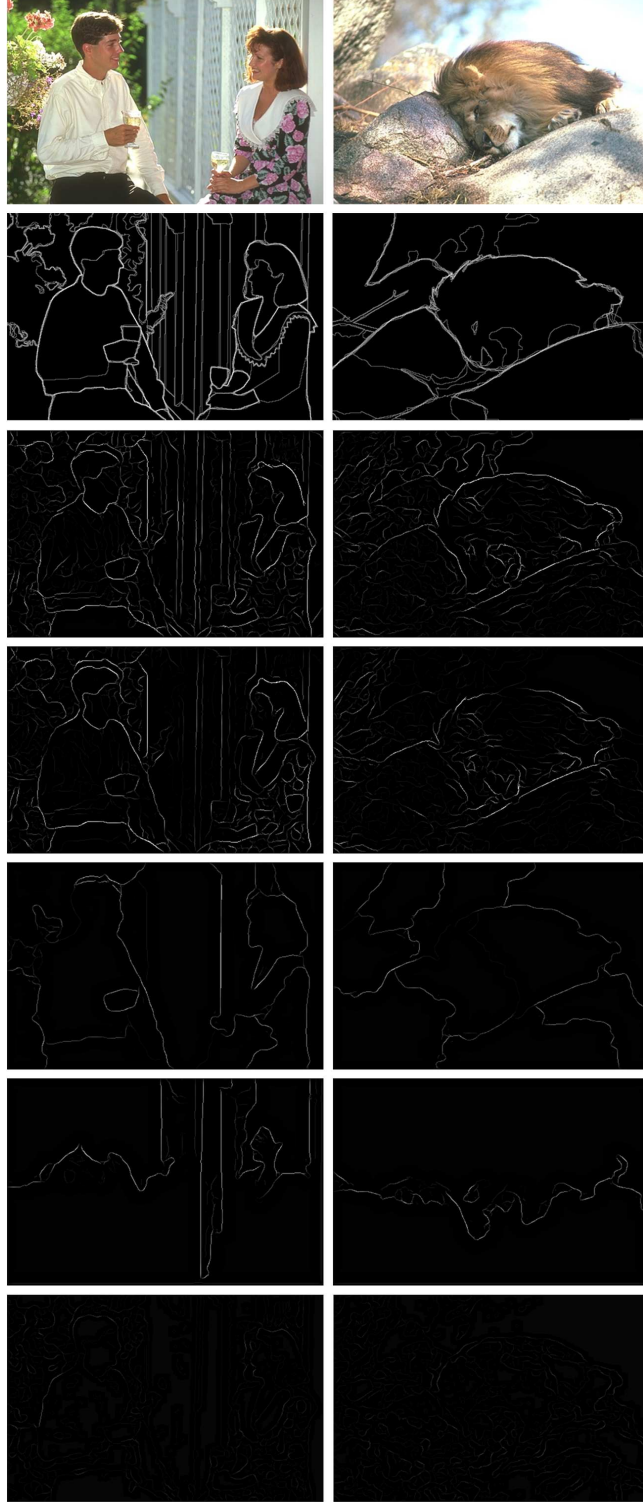


Fig. 11. Row wise top to bottom: Original image, probabilistic boundaries for human segmentation, mTch ($N_{par} = 0.95$, $\epsilon = 4$ and number of clusters 10), EM (number of clusters 10), nCuts (number of clusters 10), grBase ($\sigma = 0.8$, $k = 300$) and mShift ($h_r = 8$, $h_s = 8$).

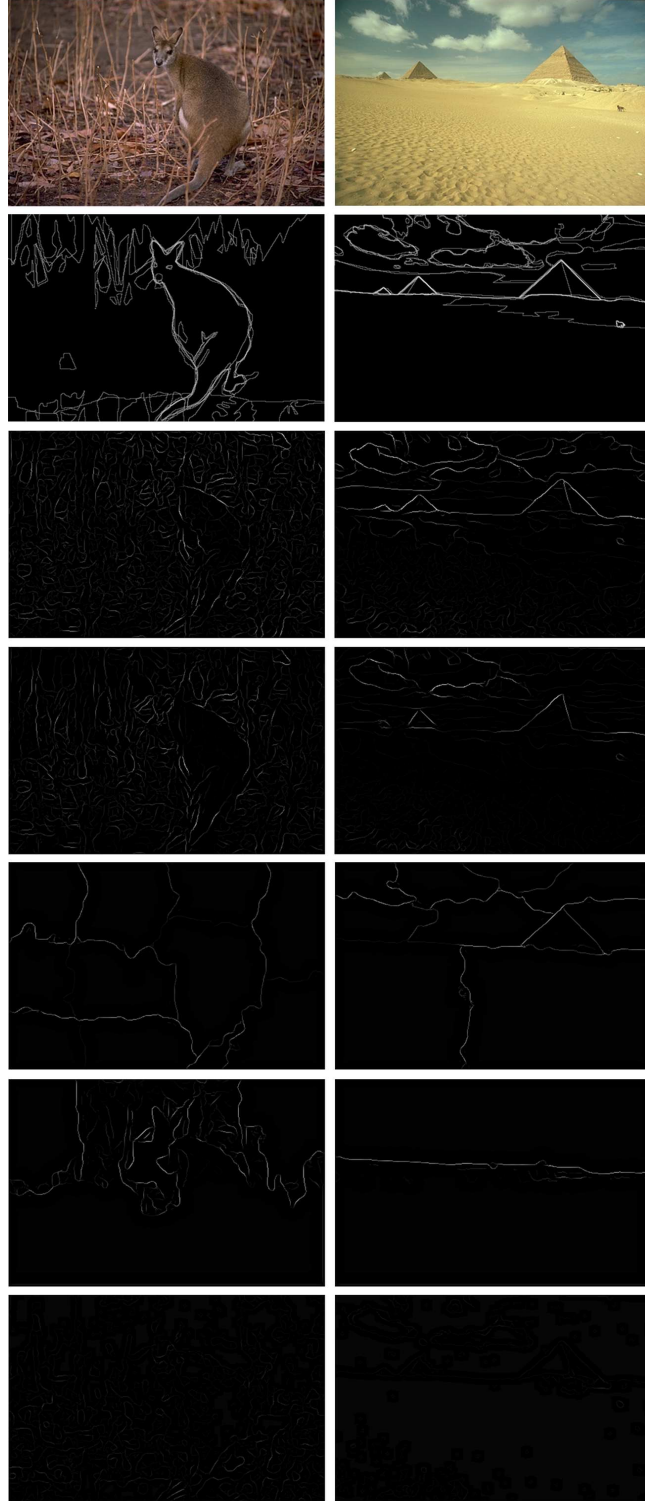


Fig. 12. Row wise top to bottom: Original image, probabilistic boundaries for human segmentation, mTch ($N_{par} = 0.95$, $\epsilon = 4$ and number of clusters 10), EM (number of clusters 10), nCuts (number of clusters 10), grBase ($\sigma = 0.8$, $k = 300$) and mShift ($h_r = 8$, $h_s = 8$).

Rank	\mathcal{F} -Score	Algorithm
0	0.79	Humans
1	0.63	Expectation Maximization (EM)
2	0.59	Multivariate Chebyshev (mTch)
3	0.58	Mean Shift (mShift)
4	0.50	Normalized Cuts (nCuts)
5	0.28	Graph Based (grBase)

Table 1. Summary table of rank of algorithms based on \mathcal{F} -score generated on 100 test images in the BSB dataset [10].

4 Experiments

To test the effectiveness of the multivariate Chebyshev algorithm (mTch), the results obtained on image segmentation were compared with some of the standard existing algorithms. The other algorithms employed were the normalized cuts (nCuts) [11], [12], edge graph based segmentation (grBase) [13], the mean shift clustering (mShift) [14] and an expectation maximization (EM) based implementation [15]. Figure 13 shows the standard peppers image that was segmented into 10 clusters for mTch, EM and nCuts. For the case of grBase ($\sigma = 0.8$, $k = 300$) and mShift ($h_r = 8$, $h_s = 8$), parameters values were specified as mentioned in literature. Matlab implementations of grBase and mShift were taken from [18] to produce the results. All algorithms were tested with a fixed parameter value on images with varying content. This does imply that results per image may not be optimized and would sound a bit unfair, but from another perspective such an experiment also suggests how robust an algorithm is against the variance in content of images, given a fixed parameter value. This outlook holds true when both the grBase and mShift algorithms in this experimental setup donot fair well on fixed parameter value for all images as compared to the proposed mTch algorithm. The \mathcal{F} -scores later generated bolster this claim.

Probabilistic boundaries were generated based on brightness and texture gradients [10] on the segmented images generated from the algorithms under study as well as the available human segmentation. The boundaries were later evaluated to find \mathcal{F} -scores. In many of these images, the segmentations based on bounded surfaces gave the good and consistent results. Table 1 shows the summary of \mathcal{F} -scores by the algorithms onthe benchmark. It states that the segementations from the bounded surfaces gave better results than some of the standard well known algorithms. The mTch works second best to EM. A reason for low performace of mTch w.r.t EM may be due the ignorance of important features like surface gradient and reflectance propertices that may add to more discriminative information than using average values of intensities at the current stage. Although the EM gives good performance many times, there are places where it does not capture the nature of surface well. One such case is the texture of the purple cloth and wrinkles in it, in the peppers image (figure 13). For more such cases, the generated results can be made available.

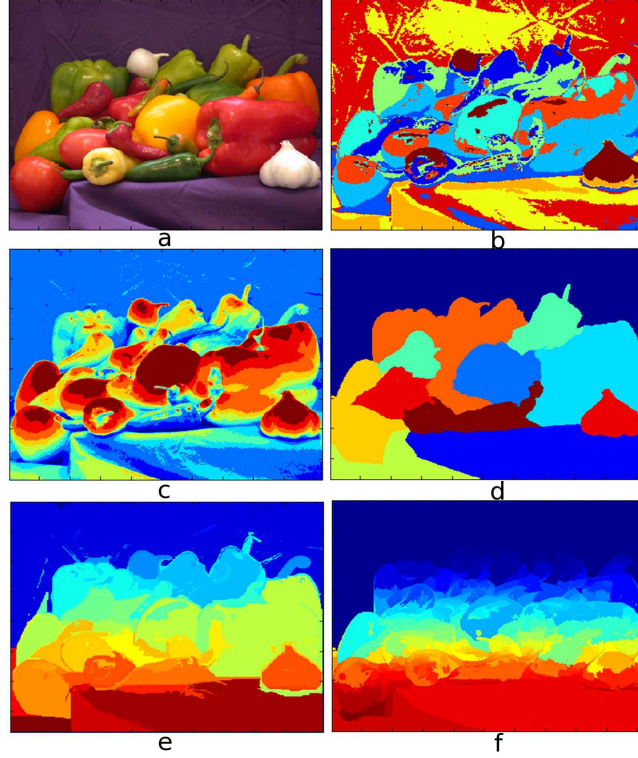


Fig. 13. (a) Original and segmented images from (b) mTch ($N_{par} = 0.95$, $\epsilon = 4$ and number of clusters 10), (c) EM (number of clusters 10) (d) nCuts (number of clusters 10) (e) grBase ($\sigma = 0.8$, $k = 300$) and (f) mShift ($h_r = 8$, $h_s = 8$).

It is widely known that the mShift yields good segments such that no local information is left behind. Too much information sometimes may not be necessary while segmentation are been compared. Figures 11 and 12 show the boundaries generated using the routines in the benchmark. Note that in these images, the mTch gave the best results. Also, at first sight it may appear that the boundaries may not have been generated well in case of mShift and grBase, but this is not the case as careful investigation does show their presence.

The grBase [13] works on the basis of a predicate that measures the evidence of a boundary on two ideas, namely: the comparison of intensity differences across boundary and the other, intensity differences among neighbourhood pixels within a region. For this a threshold function is proposed which depends on the components and their respective size. This also means that the parameter used for scaling components would be different for different images, if good segmentation results are desired in grBase. In comparison, the mTch decomposes the images into surfaces which are bounded surfaces or components that

preserve the homogeneity in texture using the image content invariant HSFC and the multivariate measure that captures the local neighbourhood interaction in the Tchebyshev inequality. Thus the constant value of parameters for mTch would give good results for different images most of the time. Results in the benchmark dataset prove the issue that for same values of parameters in grBase the segmentation results are inferior to that of mTch of a sample of 100 test images.

It must be noted that the parameter k used in [13], scales the size of the component and is not the minimum component size. Thus it is taken as a constant and no relations of it are derived with respect to the interaction present in the image. Though on similar lines, ϵ in mTch is also a parameter that defines the degree of control over size of components, but with a bound. The ϵ characterizes the size of a surface or component probabilistically, while relating to the control of the degree of texture interaction using the multivariate measure in the Tchebyshev inequality (equation 3). This probabilistic bound is the key relation between the size of component and the texture interaction within, that decomposes the image into homogeneous surfaces in $\mathcal{N}D$ by lemma 3. Clustering on a bunch of homogeneous surfaces is bound to give robust segmentation results. The price that is paid is in terms of time required to cluster the surfaces using the standard k -means.

Figures 11 and 12 show the boundaries generated using the routines in the benchmark which are later evaluated to generate \mathcal{F} -scores that determine the accuracy of the proposed algorithm as well as that of the other algorithms. Note that in these images, the mTch gave the best results. Also, at first sight it may appear that the boundaries may not have been generated well in case of mShift and grBase, but this is not the case as careful investigation does show their presence. Figure 14.a and 14.b shows the precision recall curve for (EM, mTch) and (mTch, mShift), respectively. Figure 15.a and 15.b shows the curve for (mTch, nCuts) and (mTch, grBase), respectively.

5 Discussion

The proposed method has advantages as well as disadvantages. This section gives an analysis of the intricate points of the algorithm as well as hints as to where improvements can be made.

Neighbourhood Information is implicitly considered using a HSFC traversal. The *topography of surface* is captured by treating the \mathcal{N} dimensions of the image. For RGB it would be $\mathcal{N} = 3$.

Filtering of image is not required unless absolutely necessary. The selection of surfaces and checking their validity based on variation of points in $\mathcal{N}D$ via Tchebyshev Inequality, makes the operation of filtering computationally redundant. Especially the variation or standard deviation in the $\mathcal{N}D$ points is the measure that tolerates the amount of noise that can be taken into account. If the variation is too high, then the initialized surface may be invalid for further processing due to excessive noise.

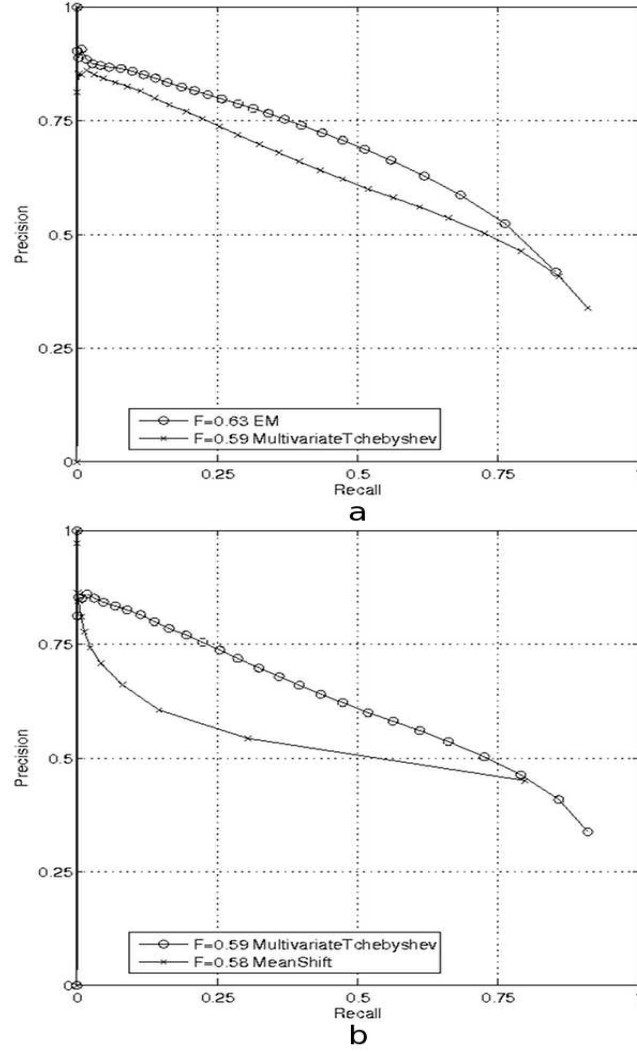


Fig. 14. Precision Recall curve for (a) EM (number of clusters 10) and mTch ($Npar = 0.95$, $\epsilon = 4$ and number of clusters 10) and (b) mTch ($Npar = 0.95$, $\epsilon = 4$ and number of clusters 10) and mShift ($h_r = 8$, $h_s = 8$).

The *Leakage problem* is that it is not known when to stop to determine the size of the area. This is tackled by determining the length of the surface via the use of the Tchebyshev's Inequality, instead of using thresholds based on image intrinsic intensity values. One aspect that affects the performance of the algorithm is the k -means clustering which may require several iterations as well as replicates in order to converge and produce clusters without getting stuck in

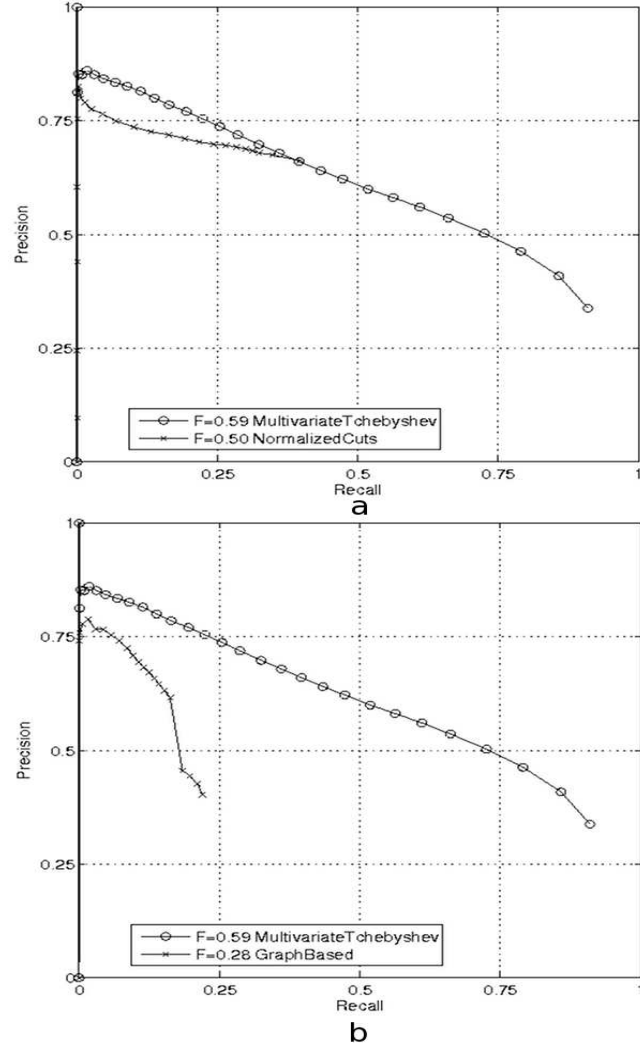


Fig. 15. Precision Recall curve for (a) mTch ($N_{par} = 0.95$, $\epsilon = 4$ and number of clusters 10) and nCuts (number of clusters 10) (b) mTch ($N_{par} = 0.95$, $\epsilon = 4$ and number of clusters 10) and grBase ($\sigma = 0.8$, $k = 300$).

some local minima. In general terms, if one is not bogged down by the intricacies of the k -means then the whole framework works well on surfaces and gives nice segmentation results on the benchmark. The k -means on pixels is slower than the k -means on the surfaces themselves as the sample size of the former is reduced by a factor of ϵ while retaining the local properties using the Hilbert space filling curve.

The current version of the algorithm also does not optimize the values of ϵ in equation 3 and $Npar$ the initialization parameter. Silhouette validation [?] could be one of the methods employed to find the best value of k clusters. The current research does not focus on the number of clusters per se. Instead of focusing as optimisation problem, $Npar$ and ϵ act as parameters of degree of control over the inclusion of points for surface initialization and size of the surface. The experiments prove that the initial model, without much tuning gives comparable results for segmentation purpose. Robust performance across images with varying context point towards the benefits of using a hybrid model that currently uses a fixed Chebyshev parameter value, a simple measure of similarity and a fixed space filling curve. Intuitively, it can be inferred that clustering on these probabilistically bounded homogeneous surfaces will be faster than clustering on pixels. This is because the homogeneous intensity values gets bundled up together and reduces the sample size of the original problem by a factor of ϵ (Chebyshev parameter). Because of its generalized framework, the proposed decomposition algorithm can find its application in areas like the generation of textures, combining information from multimodal sources as in biomedical images and processing multidimensional information on space filling curves, to name a few.

6 Conclusion

A novel hybrid model for image decomposition has been proposed. The model works on reduced image representation based on monovariate functions and processes information spread in a multidimensional framework. Initial segmentation results indicate the efficacy of the tool in terms of generalisation and robustness across images with varying content.

Acknowledgement

The author thanks the Neuroimaging Center (UMCG, Groningen, The Netherlands) for supporting this work and Dr. Charless Fowlkes, assistant professor at Computer Science Department UC Irvine (USA), for sharing the compiled Berkeley Segmentation Benchmark code for Mac intel hardware.

References

1. Sprecher, D.A.: An improvement in the superposition theorem of Kolmogorov. Journal of Mathematical Analysis and Applications, vol. 38, 208-213 (1972)
2. Sprecher, D.A., Draghici, S.: Space-filling curves and Kolmogorov superposition-based neural networks. Neural Networks, vol. 15, no. 1, 57-67 (2002)
3. Leni, P.E., Fougere, Y., Truchetet, F.: Kolmogorov Superposition Theorem and its application to multivariate function decompositions and image representation. Conference on Signal-Image Technology and Internet-Based Systems (2008)

4. Leni, P.E., Fougerolle, Y., Truchetet, F.: Kolmogorov Superposition Theorem and its application to wavelet image decomposition. *Proceedings of SPIE - Electronic Imaging*, no. 7248 (2009)
5. Igelnik, B., Parikh, N.: Kolmogorov's spline network. *IEEE transactions on neural networks*, vol. 14, no. 4, 725-733 (2003)
6. Hilbert, D.: Uber die stetige Abbildung einer Linie auf ein Flächenstück. *Math. Ann.*, vol. 38, 459-460 (1891)
7. Jagadish, H.V.: Linear clustering of objects with multiple attributes. *SIGMOD Rec.*, vol. 19, no. 2, 332-342 (1990)
8. Moon, B., Jagadish, H.V., Faloutsos, C., Saltz, J.H.: Analysis of the Clustering Properties of the Hilbert Space-Filling Curve, *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, 124-141 (2001)
9. Chen, X.: A New Generalization of Chebyshev Inequality for Random Vectors. *arXiv:math.ST*, vol. 0707, no. 0805v1, 1-5 (2007)
10. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, 416-423 July (2001)
11. Shi, J.B., Malik, J.: Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, 888-905 (2000)
12. Cour, T., Benezit F., Shi, J.: Spectral Segmentation with Multiscale Graph Decomposition. *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 1124-1131 (2005)
13. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, vol. 59, no. 2, September (2004)
14. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, 603-619 (2002)
15. Herrera, J.V.M.: EM based image segmentation. www.mathworks.com/matlabcentral/fileexchange/10956, (2006)
16. Gotsman, C., Lindenbaum, M.: On the metric properties of discrete space-filling curves. *IEEE Transactions on Image Processing*, vol. 5, no. 5, 794-797 (1996)
17. Lau, D.: Hilbert Space filling matlab code. <http://www.mathworks.com/matlabcentral/fileexchange/125> (1998)
18. Woodford, O.J., Torr, P.H.S., Reid, I.D., Fitzgibbon, A.W.: Global Stereo Reconstruction under Second Order Smoothness Priors. *IEEE Transactions in PAMI*, vol. 31, no. 12, 2115-2128 (2009)